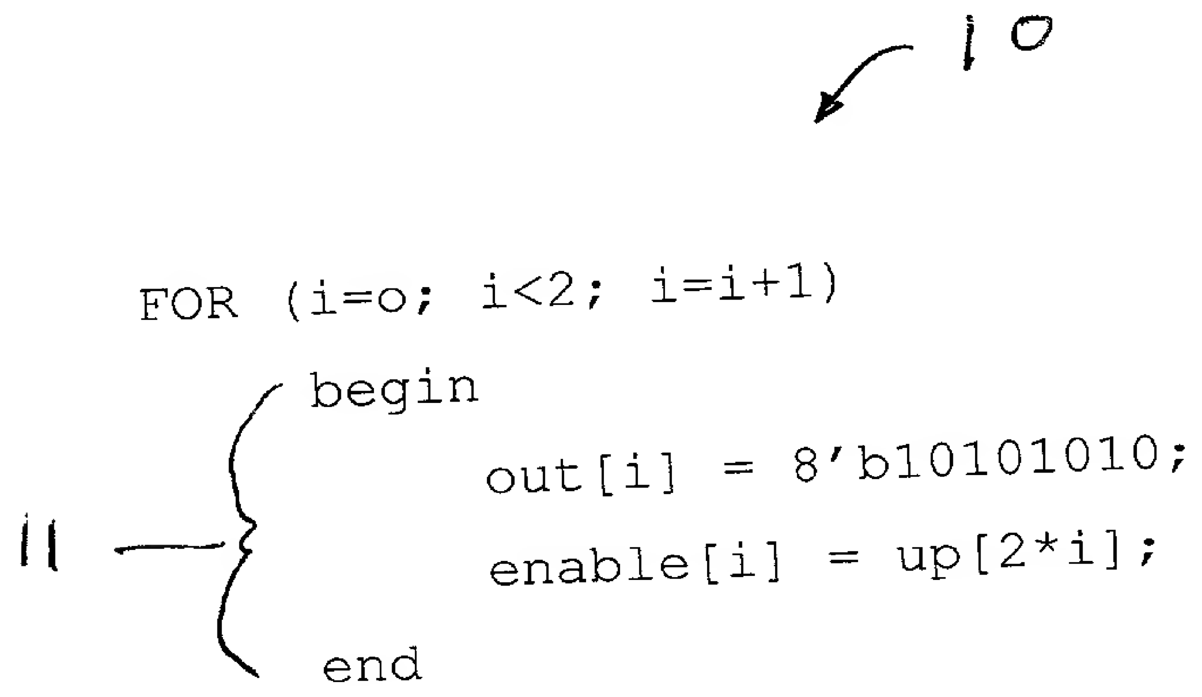


```
FOR (i=o; i<2; i=i+1)
    begin
          out[i] = 8’b10101010;
          enable[i] = up[2*i];
    end
```

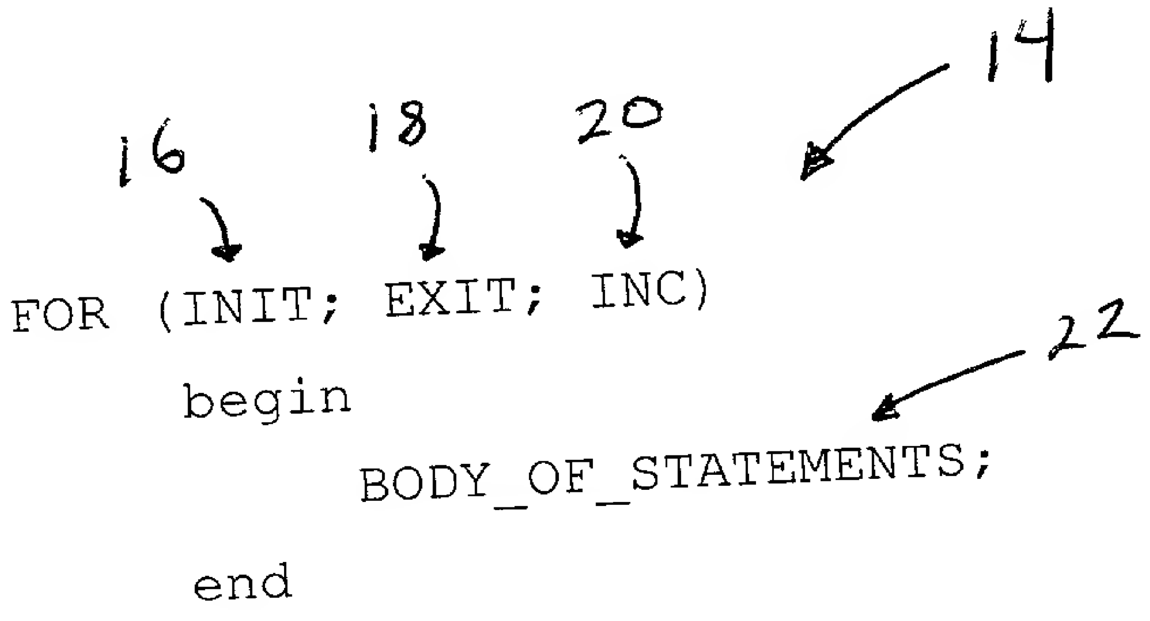
FIGURE 1

13

12

```
reg y [3:0];
WHILE (x <= y)
     begin
          fpl_bit[x+y] = mm_iru[x-y];
     end
```

FIGURE 2



```
FOR (INIT; EXIT; INC)
     begin
          BODY_OF_STATEMENTS;
     end
```

FIGURE 3

```
out[0]  = 8'b10101010;
enable[0]  = ~up[0];

out[0]  = 8'b10101010;
enable[0]  = ~up[2];
```

24

FIGURE 4

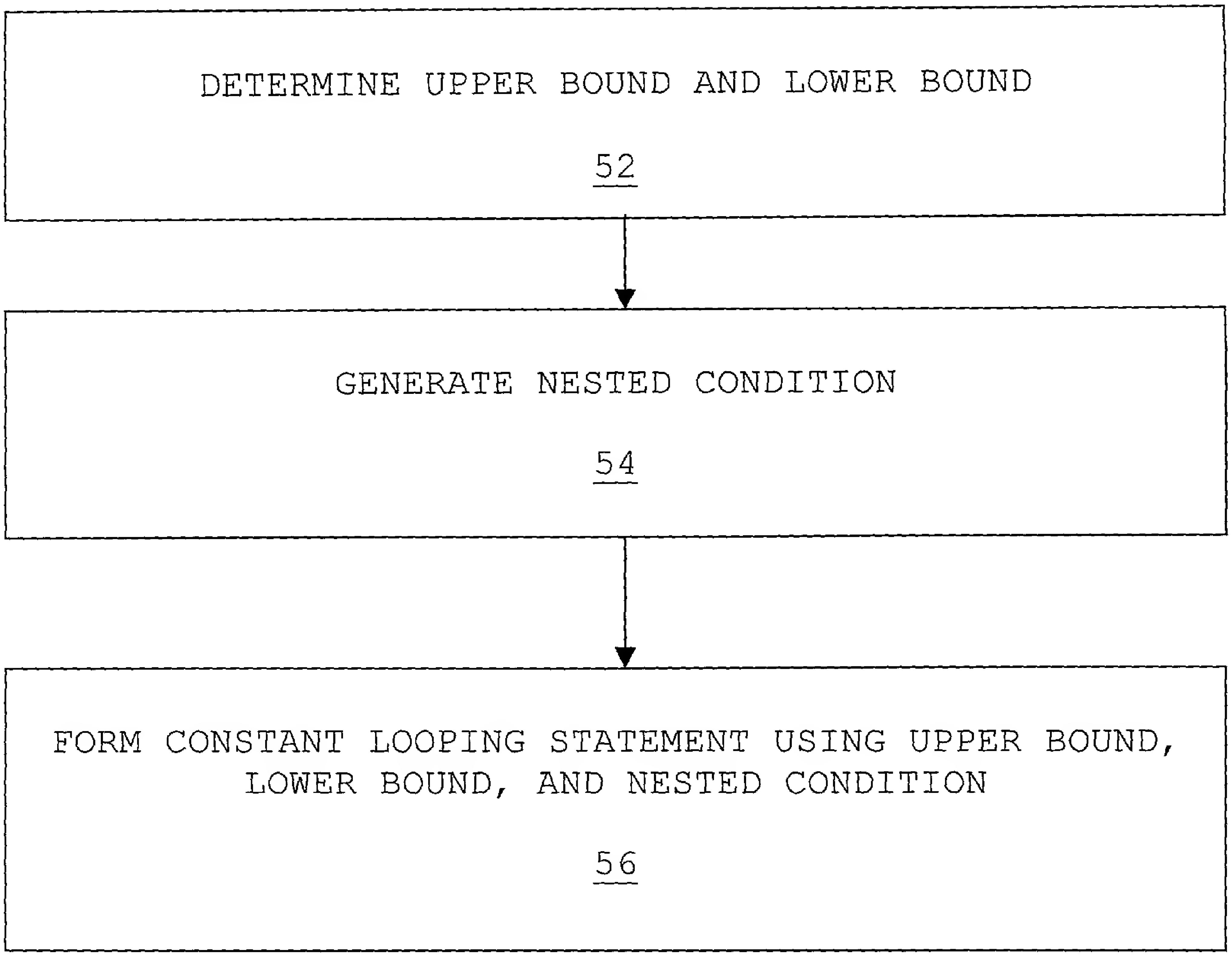
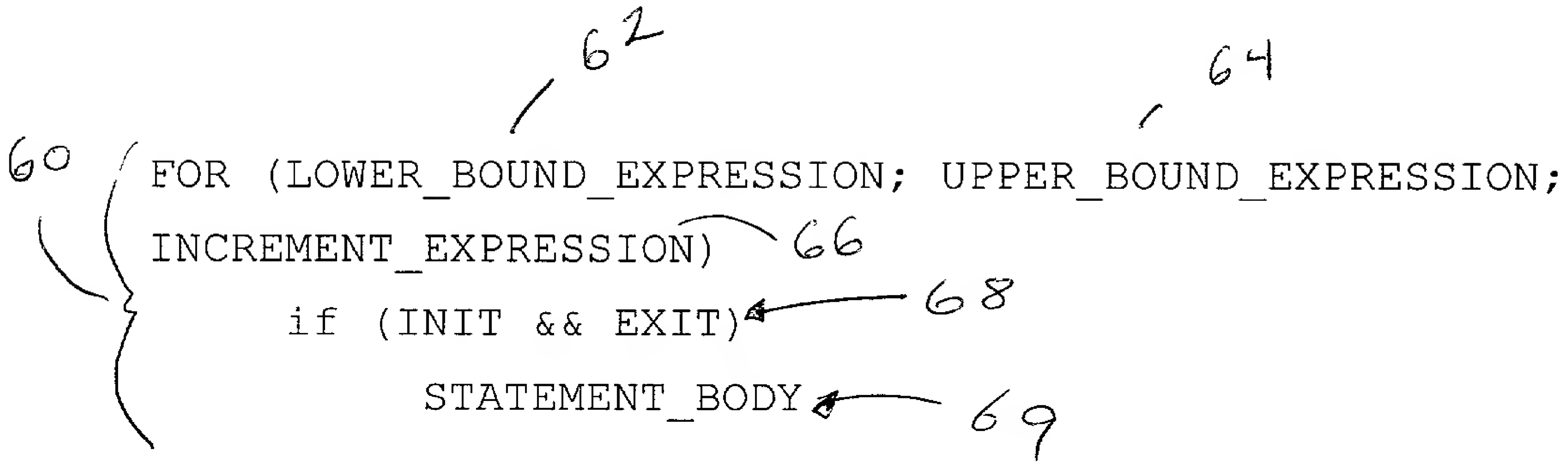
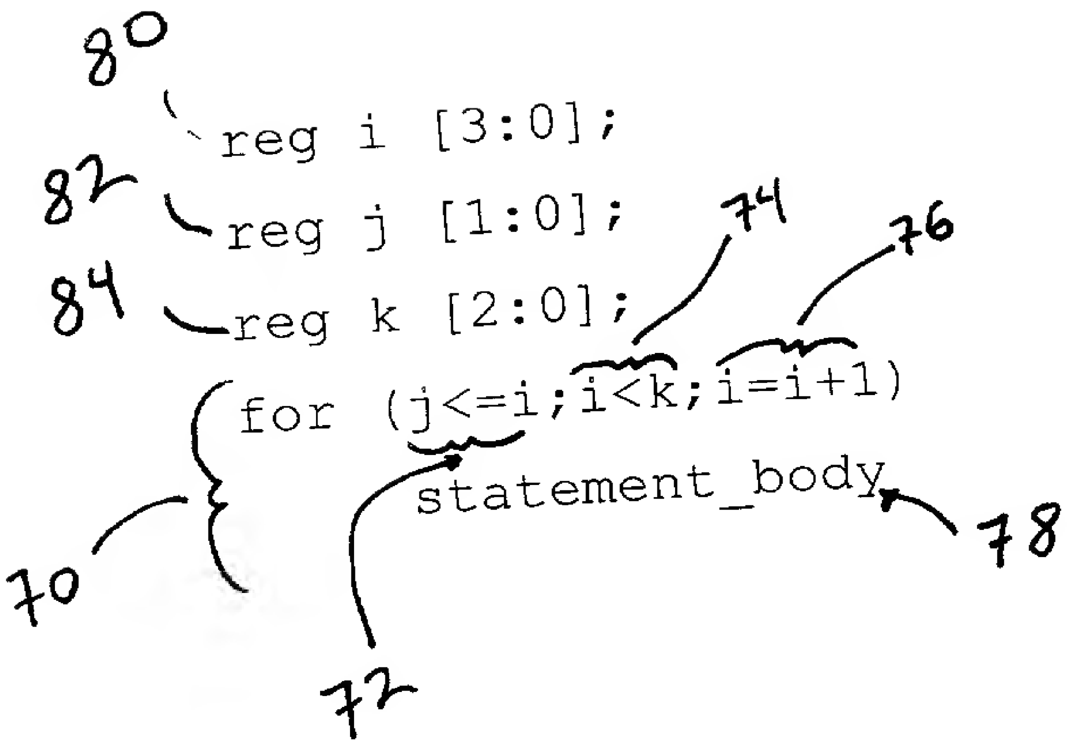

DETERMINE UPPER BOUND AND LOWER BOUND
52
GENERATE NESTED CONDITION
54
FORM CONSTANT LOOPING STATEMENT USING UPPER BOUND, LOWER BOUND, AND NESTED CONDITION
56

FIGURE 5



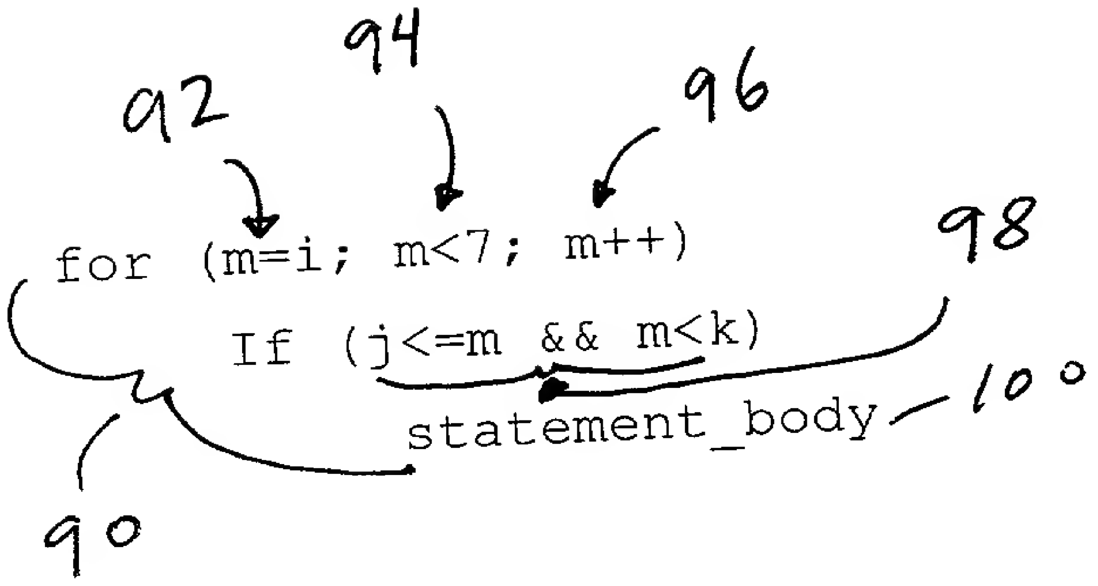
```
FOR (LOWER_BOUND_EXPRESSION; UPPER_BOUND_EXPRESSION;
INCREMENT_EXPRESSION)
     if (INIT && EXIT)
          STATEMENT_BODY
```

FIGURE 6



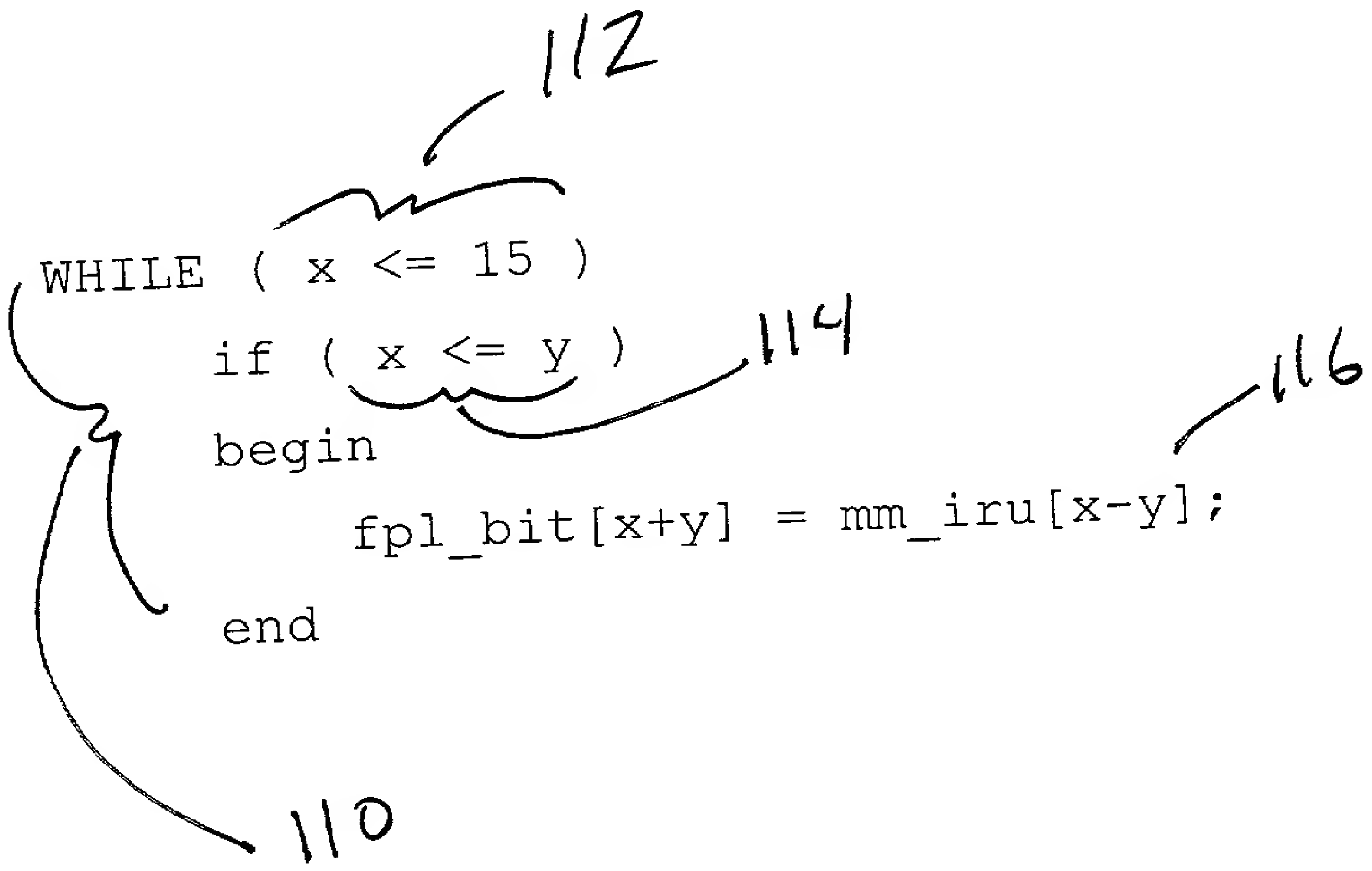
```
reg i [3:0];
reg j [1:0];
reg k [2:0];
for (j<=i;i<k;i=i+1)
   statement_body
```

FIGURE 7



```
for (m=i; m<7; m++)
        If (j<=m && m<k)
                statement_body
```

FIGURE 8



```
WHILE ( x <= 15 )
     if ( x <= y )
     begin
         fpl_bit[x+y] = mm_iru[x-y];
     end
```

FIGURE 9